Name: Sahib Bajwa
ID: 107553096

CSCI 3104, Algorithms **Profs. Chen & Grochow**
Final Exam S18–S20 **Spring 2020, CU-Boulder**

Instructions: This quiz is open book and open note. You **may** post clarification questions to Piazza, with the understanding that you may not receive an answer in time and posting does count towards your time limit. Questions posted to Piazza **must be posted as PRIVATE QUESTIONS.** Other use of the internet, including searching for answers or posting to sites like Chegg, is strictly prohibited. Violations of these are grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show and justify all work to receive full credit.**

TIMING: If you are not attempting all the standards in a given quiz, please only use the ordinary amount of time for the number of standards you attempt. For example, if you are only attempting one standard on a 4-standard quiz, please only use 30 min (or 38 for 1.5x, 45 for 2x).

YOU MUST SIGN THE HONOR PLEDGE. Your quiz will otherwise not be graded.
Honor Pledge: On my honor, I have not used any outside resources (other than my notes and book), nor have I given any help to anyone completing this assignment.

Your Name: Sahib Bajwa

Quicklinks: 18 19 20

---

18. **Standard 18.** Suppose we have n stairs to climb. You may choose to jump up either $1, 3,$ or 5 stairs each time. Your goal is to find the **minimum number of jumps** to climb the stairs. Note that your starting position is on the ground floor and not on the first stair.

    Is there a clear recursive structure in the problem that would be useful in designing an effective dynamic programming algorithm? That is, is dynamic programming a useful algorithmic technique for this problem? Clearly justify your answer.

    (BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR STANDARD 18. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)

Name: Sahib Bajwa

ID: 107553096

CSCI 3104, Algorithms **Profs. Chen & Grochow**

Final Exam S18–S20 **Spring 2020, CU-Boulder**

19. **Standard 19.** Consider a set of coin denominations in cents $\{d_1, d_2, \cdots, d_k\}$ with $d_1 = 1 < d_2 < \cdots < d_k$, and suppose you want to make change for n cents. Your goal is to find the number of ways to make the change. Note that the order of the coins does not matter, e.g., if the denominations are standard US coin denominations and we are making change for 6 cents, then 1 penny + 1 nickel and 1 nickel + 1 penny count as *one* way of making change, not two distinct ways. (There are only two ways to make change for 6 cents using US coin denominations: 6 pennies, or 1 penny + 1 nickel.)

    Define $X(i, m)$ to be the number of ways to make change for m cents using the coins of first i denominations $\{d_1, d_2, \cdots, d_i\}$, $i \leq k$. Write down a recurrence for counting the number of ways to make change, and justify it.

    (BEGIN YOUR ANSWER ON THE NEXT PAGE.)

Case 1: If we choose not to include coin k, then remove d_k from our set of coins (our coins may be drawn from all other coin possibilities).

Case 2: Suppose instead we choose to include coin k, then we create a new subproblem of the remaining amount and solve for that. We run the subproblem as its own new problem, $X(i, m)$ with the value of k subtracted from m (the cents we are trying to make).
Note: Since we can use the same coin multiple times, we do not remove it from our list of possible next coins.

The set of configurations from Case 1 and the set of configurations from Case 2 share no common configurations, so we can combine them. We get:
$X(i, m) = X(i - d_k, m) + X(i, m - k)$

Note: We can replace d_k and k in this recurrence with 1 (similar to the problem set). When doing this, we get:
$X(i, m) = X(i - 1, m) + X(i, m - 1)$

Our base cases are:
0 if the cents target, m, has been met.
1 if there is only one type of coin denomination left in the set of $\{d_1, d_2, \cdots, d_i\}$ (we can only add 1 penny from now on).

Name: Sahib Bajwa
ID: 107553096

CSCI 3104, Algorithms | **Profs. Chen & Grochow**
Final Exam S18–S20 | **Spring 2020, CU-Boulder**

20. **Standard 20.** Consider the weighed interval scheduling problem for the job list

$$A = [(1,3;4),(2,4;5),(3,5;2),(3,6;3),(2,7;5)]$$

of (start time, finish time; value) triples. Fill in the values of the following DP table, where $OPT(i)$ is the value of optimal solution to the problem consisting of the first i jobs.

triple	$OPT(i)$
(1, 3; 4)	
(2, 4; 5)	
(3, 5; 2)	
(3, 6; 3)	
(2, 7; 5)	

(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR STANDARD 20. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)